Crossing cuts polygonal puzzles: Models and Solvers

Peleg Harel
pelegh@post.bgu.ac.il

Ohad Ben-Shahar
ben-shahar@cs.bgu.ac.il

Abstract

Jigsaw puzzle solving, the problem of constructing a coherent whole from a set of non-overlapping unordered fragments, is fundamental to numerous applications, and yet most of the literature has focused thus far on less realistic puzzles whose pieces are identical squares. Here we formalize a new type of jigsaw puzzle where the pieces are general convex polygons generated by cutting through a global polygonal shape with an arbitrary number of straight cuts. We analyze the theoretical properties of such puzzles, including the inherent challenges in solving them once pieces are contaminated with geometrical noise. To cope with such difficulties and obtain tractable solutions, we abstract the problem as a multi-body spring-mass dynamical system endowed with hierarchical loop constraints and a layered reconstruction process that is guided by the pictorial content of the pieces. We define evaluation metrics and present experimental results on both apictorial and pictorial puzzles to indicate that they are solvable completely automatically.

1. Introduction and related work

Originally a children's game, the jigsaw puzzle problem aims to reconstruct a coherent whole from unordered fragments by matching their shape and/or their visual content. With countless real-world applications in various domains (e.g., [19, 23, 26]), it was first introduced as a computational task in 1964 by Freeman and Garder [14] and later evolved into a flourishing field. The focus in the computational literature has however shifted from puzzles of arbitrarily shaped pieces to puzzles of *square* pieces where geometry plays no role and the pictorial data is *the only* source of information used for the reconstruction. While starting modestly, the suggested solvers evolved over time and although no guarantees can be provided, contemporary methods can solve *square* jigsaw puzzles of virtually any practical size (e.g., [37]). At the same time, puzzles of fragments of unconstrained (or less constrained) shape, either pictorial (e.g., [17, 22]) or apictorial (e.g., [14, 20, 32]), were researched considerably less.

Since the problem of puzzle-solving was proved NP-complete [11], much of the literature on the topic has focused on devising heuristics that facilitate solutions in many (though not necessarily all) cases, including large scale puzzles of various types. Broadly speaking, the types of puzzles addressed in the prior art can be categorized into four classes based on the geometric properties of their pieces: **Commercial toy puzzles** [5, 8, 10, 16, 21, 29, 44, 45, 47], **Square jigsaw puzzles** [1–4, 7, 13, 15, 27, 28, 30, 31, 33, 36–40, 42, 46, 48, 50], **Partially constrained modelled puzzles** [17], and **Unrestricted puzzles** [14, 20, 22, 24, 25, 32, 34, 43, 49]. This classification also implies that matching puzzle pieces next to each other during puzzle reconstruction may be done differently in each class. In particular, square jigsaw puzzles *must be* pictorial in order to escape trivial setting. For sake of space we next discuss briefly only those puzzle types that are more relevant to our own work in this paper.

Partially constrained modelled puzzles are puzzles with formal yet looser geometric properties than the square jigsaw and the commercial toy puzzles. To our best knowledge, the only prior work introducing this type of puzzles is the "brick wall" model [17] that extends square jigsaw puzzles to *rectangular* pieces of different lengths. The reconstructed puzzle may thus have multiple neighbors for each piece, and the solver must allow for arbitrary offsets between neighbors, thus greatly increasing the (already exponentially large) search space. While solving *apictorial* brick wall puzzles may be possible, it is clearly NP-hard. *Pictorial* brick walls thus leverage the pictorial information to evaluate only a subset of offsets between possible neighbors. Gur and Ben-Shahar [17] thus proposed a greedy algorithm by endowing previous greedy techniques with shifting of pieces based on various compatibility measures and avoiding overlaps due to non-optimal offsets determination.

Unrestricted puzzles are puzzles that do not have formal constraints or generation model and thus their pieces can be fragments of arbitrary shapes. In general, the correct reconstruction of such puzzles can be described as a general planar adjacency graph of arbitrary maximal degree. In such puzzles, pieces can be matched to arbitrary number of neighbors abutting arbitrary section of their boundary. Additional complexity can arise form the description of the piece boundary itself. Somewhat unexpectedly, the very first work on computational puzzle solving [14] belongs to this class and like subsequent apictorial puzzle solvers [20, 32] it uses curve matching to find potential matching pieces. Freeman and Garder [14] introduced a solver capable of dealing with a large variety of piece shapes and junction types, and employed a chain encoding scheme to match piece boundaries. The solver tried to reconstruct around junctions, thus seeking neighbors with loopy consensus, perhaps leading the way to the future use of various loopy constraints in the field (e.g., [31, 38]). However, their exhaustive DFS-like search that backtracks on errors was feasible only because of the very small scale problems considered (9 piece puzzles) and did not permit later experimental comparison to contemporary contributions. 40 years later, Kong and Kimia [20] used a coarse-to-fine approach to curve matching and a greedy merging of piece triplets and backtracking upon spatial overlap. In their paper, three puzzles were reconstructed with up to 25 pieces, while the data used for testing consisted of pieces resembling convex polygons of few edges.

Solvers for unrestricted *pictorial* puzzles [22, 24, 25, 34, 43, 49] use the pictorial content, as well as geometrical boundary, to position the pieces. Sağıroğlu and Erçil [34] extrapolated the pictorial content of each piece in a band around it in order to score the match to prospective neighbors. The reconstruction itself was done in a greedy fashion, starting from a random configuration and improving the global score one a piece at a time. Local minima were handled by repeating the process from several random seed configurations. The experimental evaluation was however limited to assemblies of just 21 pieces.

A different approach altogether was recently introduced by Le et al. [22] who computed piecewise affinity with a convolutional neural network that utilizes both boundary shape and pictorial data. Their solver was tested successfully on puzzles of up to 400 pieces of roughly perturbed rectangular shape.

2. The "Crossing cuts" puzzle

Recall that the pieces of the square jigsaw puzzle are all identical in shape, a setup that drives all reconstruction decisions to the pictorial realm. However, real-world puzzles usually have pieces of a more general form [35], leading to a different set of challenges. In this paper we formulate a new class of puzzles that (unlike square puzzles) is general enough for more real world applications and yet (unlike unrestricted puzzles) formal enough for rigorous analysis and exploration.

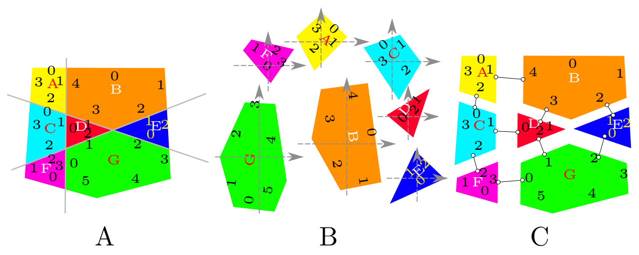


Figure 1: The elements of a crossing cuts puzzle. **A:** The puzzle is created by cutting a convex polygon using multiple (here 3) straight cuts. **B:** The puzzle problem constitutes of an unordered and arbitrarily transformed set of pieces. Each piece (here $\{p_A, p_B, p_C, p_D, p_E, p_F\}$) is represented by its vertices and edges in some arbitrary Euclidean coordinate system (which conveniently may be centered at the center of mass). Note that different pieces may vary vastly in size (e.g., compare pieces p_B and p_D.) **C:** The mating graph matches pairs of edges of two different pieces. In our case it includes $\{\{e_A^1, e_B^4\}, \{e_A^2, e_C^0\}, \{e_B^2, e_E^1\}, \{e_B^3, e_D^1\}, \{e_B^3, e_D^1\}, \{e_C^1, e_D^0\}, \{e_C^2, e_F^2\}, \{e_D^2, e_G^1\}, \{e_E^0, e_G^2\}, \{e_F^3, e_G^0\}\}$. Note that pieces end up having different number of edges and thus different number of neighbors.

The crossing cuts puzzle is created by cutting through a convex polygon with $a \in \mathcal{N}$ arbitrary (random) straight cuts $Cuts = \{c_1, \dots c_a\}$, as illustrated in Fig. 1A. The pieces of such puzzles are thus convex polygons, where every piece (except border pieces) has a single neighbor along each of its edges. Once shuffled (Fig. 1B), the challenge is to reconstruct the original polygon again. Here we show apictorial puzzle but later on we add pictorial content as well.

A crossing cuts *puzzle* (in the sense of the riddle to solve) is a representation of the unordered puzzle pieces after the complete polygon was cut. Formally, let $P = \{p_1, \dots p_n\}$ be a set of *pieces*, where each piece p_i is a convex polygon of $N_i \geq 3$ vertices $p_i = \left\{\vec{v}_i^{\,1}, \vec{v}_i^{\,2}, \dots, \vec{v}_i^{\,N_i}\right\}$. As a convention, we order these vertices clockwise around the polygon's center of mass. Correspondingly we label the piece edges between these consecutive vertices by

$$E_i = \left\{e_i^1, e_i^2, \dots, e_i^{N_i}\right\} = \left\{(\vec{v}_i^{\,1}, \vec{v}_i^{\,2}), (\vec{v}_i^{\,2}, \vec{v}_i^{\,3}), \dots, (\vec{v}_i^{\,N_i}, \vec{v}_i^{\,1})\right\} .$$

A *solution* to a crossing cuts puzzle, unlike the puzzle itself, essentially requires to position each piece in its "correct" position relative to all other pieces, and while this only requires the determination of a Euclidean transformation (position and rotation) for each piece, in practice this will first require to resolve the "correct" neighborhood relationships between the pieces. We represent such immediate neighborhood relationships with *matings* links [14], while the connected pieces themselves are called *neighboring pieces* and the matched edges are called *mates*. Note that in the ideal case, when no noise is present, a mating represents two *overlapping* mates with *identical* lengths.

Unlike in square piece (and also commercial toy) jigsaw puzzles, which have a constant number of neighbors for each piece (except boundary pieces), the number of matings a piece can have in our mating graph is arbitrary. Moreover, the number of possible Euclidean configurations (translation and rotation) of the pieces of crossing cut puzzles adds additional complexity, since it is infinite and selected from a continuous range. At the same time, the geometry of the pieces provides more information that is not present in the square jigsaw problem, and may facilitate reconstruction algorithms that rely only on their shape. An algorithm to obtain a solution thus needs to cope with these properties and determine both

i. the pairwise matings $M = \left\{m_1, \dots m_{|M|}\right\}$ of all pieces, i.e., all unordered pairs of edges $m_q = \{e_i^j, e_k^l\}$ of two *different* pieces that should be matched (and in an ideal setting, truly overlap) in order to reconstruct the puzzle, and

ii. the 2D Euclidean transformation of each piece p_i, from its given input representation to the one in the reconstructed puzzle. The transformation of piece p_i involves a translation $t_i \in \mathcal{R}^2$ and a rotation $R_i \in \mathcal{S}^1$. With the rotation typically represented by an orthonormal matrix $R_i \in \mathcal{R}^{2\times 2}$, the pose of a piece in the reconstructed puzzle is $p'_i = \left\{R_i \cdot \overrightarrow{v}_i^1 + \overrightarrow{t}_i, R_i \cdot \overrightarrow{v}_i^2 + \overrightarrow{t}_i, \dots, R_i \cdot \overrightarrow{v}_i^{N_i} + \overrightarrow{t}_i\right\}$.

Fig. 1 illustrates both the puzzle and the aspects of its solution as just discussed.

3. Mating constraints and greedy solver

Assuming no noise, idealized infinite precision in the representation of the geometrical objects, and uniform random distribution of the crossing cuts themselves, it is immediate to observe that the probability of (1) more than two crossing cuts to meet at a point and (2) having more than two piece edges with *identical* lengths, is nil in both cases. These properties of the *generic* (i.e., non accidental) puzzle entail two key constraints for the formation of plausible matings constraints:

C_1: **Mate length constraint:** Since plausible matings match complete edges, their corresponding mates must have the very same length.

C_2: **Mate angle constraint:** Since plausible mates have vertices emerging from just 2 crossing cuts, it follows that the two pairs of adjacent angles of the neighboring pieces must complete to π, i.e., be supplementary (see Fig. 2B).

In the following we will refer to the mating constraints also as predicates, i.e., $\forall i \in \{1,2\} \quad C_i\left(e_i^j, e_k^l\right) = \text{True}$ iff e_i^j and e_k^l satisfy constraint C_i.

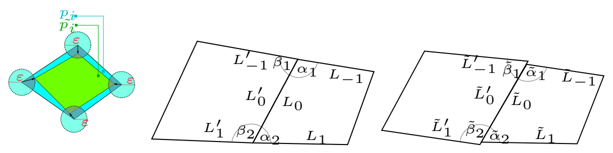
Clearly, the constraints just outlined entail a simple, *greedy*, yet sound and complete baseline solver that starts from an initial (random) piece edge and greedily places the *only* (i.e., single) matching unassigned edge of some other piece next to it (i.e., while satisfying C_1 and C_2) while setting the proper Euclidean transformation of the corresponding piece accordingly.

4. Noisy crossing cuts puzzles

Real life crossing cuts puzzles (or geometric puzzles in general) are never perfect and may incorporate deformed pieces. Such noise can be modelled in many different ways, though one particular appealing is material degradation, and thus piece erosion, a process clearly relevant for applications involving physical pieces (e.g. in archaeology). To model material degradation in our crossing cuts context, we preserve the number of vertices of each piece, but shift (i.e., collapse) each of them *inward* by a random distance that is distributed uniformly in a given range. Formally, vertex $\overrightarrow{v}_i^j$ of piece p_i is perturbed inwards by a distance $\overrightarrow{\epsilon}_i^j$ that is bounded relative to the puzzle diameter D (distance between furthest vertices). Let ξ be that bound, that sets the absolute noise bound at $\varepsilon = \xi \cdot D$. An original piece $p_i = \left\{\overrightarrow{v}_i^1, \dots \overrightarrow{v}_i^{N_i}\right\}$ thus becomes the following *ε-noisy* piece $\tilde{p}_i = \left\{\overrightarrow{v}_i^1 + \overrightarrow{\epsilon}_i^1, \dots, \overrightarrow{v}_i^{N_i} + \overrightarrow{\epsilon}_i^{N_i}\right\}$ where $\left\|\overrightarrow{\epsilon}_i^j\right\| \sim \mathrm{U}(0,\varepsilon)$, $\measuredangle\overrightarrow{\epsilon}_i^j \sim \mathrm{U}\left(\measuredangle\overrightarrow{e}_i^j, \measuredangle\overrightarrow{e}_i^{j+1}\right)$ and $\measuredangle e_i^j, \measuredangle e_i^{j+1}$ are the angles of the piece edges leaving $\overrightarrow{v}_i^j$ towards the nearby vertices. Fig. 2A illustrates how such noise could affect the shape of a quadrilateral (4-side) piece. Naturally, the incorporation of noise affects the validity of the mating constraints and C_1 and C_2 must be revised to $\tilde{C}_1, \tilde{C}_2$:

$\tilde{C}_1$: If $\tilde{e}$ and $\tilde{e}'$ are two corresponding ε-noisy mates, their respective lengths $\tilde{L}$ and $\tilde{L}'$ should satisfy $\left|\tilde{L} - \tilde{L}'\right| \leq 4\varepsilon$. The maximum error ($4\varepsilon$) can occur when one of the edges is shortened by 2ε and the other is lengthened by 2ε due to the noise.

$\tilde{C}_2$: Let L_{-1}, L_0, L_1 and L'_{-1}, L'_0, L'_1 be the lengths of the edges *before*, *at*, and *after* the mating (e, e'), respectively (see Fig. 2B). Let α_1, β_1 and α_2, β_2 be the pairs of supplementary angles that these two



p_j
p_i
ε
L'_{-1} β_1 α_1 L_{-1}
L'_0 L_0
L'_1 β_2 α_2 L_1
$\tilde{L}'_{-1}$ $\tilde{\beta}_1$ $\tilde{\alpha}_1$ $\tilde{L}_{-1}$
$\tilde{L}'_0$ $\tilde{L}_0$
$\tilde{L}'_1$ $\tilde{\beta}_2$ $\tilde{\alpha}_2$ $\tilde{L}_1$
A
B
C

Figure 2: The effect of noise on the geometric constraint. **A:** Each of the vertices of a piece p_i is collapsed inwards along a uniformly distributed direction and as far as a uniformly distributed distance to create the ε-noisy piece $\tilde{p}_i$. **B:** Without noise, angles must comply to the original constraint $\alpha_1 + \beta_1 = \alpha_2 + \beta_2 = \pi$. **C:** After applying the noise the ε-noisy angles are affected by the change in orientation in all edges that meet at both vertices of the mating, to result in the bound in the text.

mates e, e' make with their adjacent edges at their vertices, as illustrated in Fig. 2B. Let $\tilde{\alpha}_i, \tilde{\beta}_i$ $i \in \{1, 2\}$ be the angles corresponding to α_i, β_i after applying the noise (as in Fig. 2C). The revised mate angle constraint becomes:

$$\begin{aligned}
|\pi - \tilde{\alpha}_1 - \tilde{\beta}_1| \leq\ & \Delta\Theta_e(\tilde{L}_0 - 2\varepsilon, \varepsilon) + \Delta\Theta_e(\tilde{L}_{-1} - 2\varepsilon, \varepsilon) \\
& + \Delta\Theta_e(\tilde{L}'_0 - 2\varepsilon, \varepsilon) + \Delta\Theta_e(\tilde{L}'_{-1} - 2\varepsilon, \varepsilon) \\
|\pi - \tilde{\alpha}_2 - \tilde{\beta}_2| \leq\ & \Delta\Theta_e(\tilde{L}_0 - 2\varepsilon, \varepsilon) + \Delta\Theta_e(\tilde{L}_1 - 2\varepsilon, \varepsilon) \\
& + \Delta\Theta_e(\tilde{L}'_0 - 2\varepsilon, \varepsilon) + \Delta\Theta_e(\tilde{L}'_1 - 2\varepsilon, \varepsilon)
\end{aligned}$$

$$\text{where } \Delta\Theta_e(L, \varepsilon) = \begin{cases} \arcsin\left(\frac{\varepsilon}{L-\varepsilon}\right) & L > 2\varepsilon \\ \infty & L \leq 2\varepsilon \end{cases}.$$

Please refer to the Supp for the proofs. As with the "clean" case, we refer to the noisy mating constraints as predicates, i.e., $\forall i \in \{1, 2\}$ $\tilde{C}_i\left(e_i^j, e_k^l\right) = \text{True}$ iff e_i^j and e_k^l satisfy constraint $\tilde{C}_i$. Clearly, with $\tilde{C}_1, \tilde{C}_2$ replacing C_1, C_2, the number of potential mates increases drastically and far from uniqueness, entailing drastic implications on a reconstruction algorithm.

5. Puzzle properties

One of the advantages of partially constrained modelled puzzles (cf. Sec. 1) is the better ability to analyze their properties. Since crossing cuts puzzles are results of a stochastic process, their properties are typically probabilistic, but nevertheless can provide insights on both the problem itself and about potential solutions (or limitations thereof). Here we explore such properties either analytically (in Table 1) or empirically (in Fig. 3), while for space considerations the proofs are deferred to the Supp. Note that these particular results assume that the global puzzle shape is a unit circle (or a polygonal approximation thereof), whose symmetry simplifies some of the analytical analyses.

6. Noisy puzzle reconstruction

At first sight one may wish to extend the initial greedy algorithm from Sec. 3 to find matings while using the relaxed "noisy" constraints ($\tilde{C}_1$ and $\tilde{C}_2$), and employ backtracking upon piece collisions. However, the expected number of possible mates per edge (c.f Sec. 5 and the Supp) clearly makes this naive extension intractable. Moreover, under noise it is unclear what is the desired position (i.e., Euclidean transformation) of each piece, or how to compute it in the first place, even if the mating relationships are set correctly. As a result, even the violations that entail backtracking in a possible algorithmic extension are ill-defined.

Table 1: Analytical properties of crossing cuts puzzles. a is the number of cuts. ξ is the level of noise relative to the diameter.

Property	**Result**
Expected cut length	$\frac{4}{\pi}$
Expected Number of cut intersections	$\frac{a(a-1)}{6}$
Expected number of edges	$\frac{a^2+2a}{3}$
Expected average edge length	$\frac{12}{\pi(a+2)}$
Noise relative to average edge length	$\frac{4 \cdot \xi \cdot \pi(a+2)}{12}$
Maximum number of pieces	$\frac{a^2}{2} + \frac{a}{2} + 1$
Expected number of pieces	$\frac{a^2}{6} + \frac{5a}{6} + 1$

To address these difficulties we approach the problem in stages, and in particular, we begin with the simpler problem of solving the puzzle *when the correct matings are given also*. More concretely, we first suggest a solution to this sub-problem by representing it as a *multi-body spring-mass system* where energy minimization is sought while the springs apply elastic forces between corresponding vertices. The solutions obtained this way are then used as scores for searching and determining the correct matings while incorporating a hierarchical (and progressively growing) set of circular constraints among adjacent pieces.

6.1. Solving noisy puzzles with *known* matings

Let $P = \{p_1, p_2, \ldots, p_n\}$ be the set of pieces and let $M = \{m_1, \ldots, m_{|M|}\}$ be the set of pairwise matings

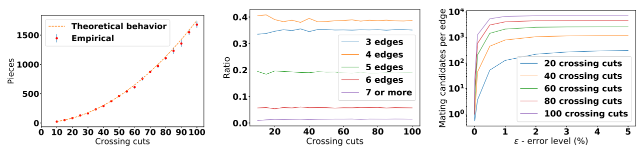


Figure 3: Selected empirical properties. **A:** Number of puzzle pieces, compared to the theoretical behavior (Table 1). Error bars are ±1 SE from 30 repetitions. **B:** Expected ratios of pieces with a particular number of edges as a function of the number of crossing cuts. Note how quadrilaterals are always the majority, followed closely by triangular pieces and the less frequent pentagons. These three classes of polygons quickly converge to account for approximately 95% of all pieces, which remain invariant to the number of cuts. **C:** The average number of potential mates as a function of noise level. The rapid growth indicates the harmful effect of noise and the need for additional *pictorial* constraints (see Sec. 7).

between their edges (i.e., $m_q = \{e_i^j, e_k^l\}$). We seek a computational scheme that obeys the given matings and places the pieces in some "optimal" or "good" way next to each other. Intuitively, we would like to do so in a way that minimizes the total L_2 displacement error between corresponding mating vertices, i.e., to find the Euclidean transformations $(R_i, \vec{t}_i)$ that satisfy

$$\underset{(R_i, \vec{t}_i)}{\operatorname{argmin}} \ \frac{1}{2} \sum_{(\vec{v}_i^{\,j}, \vec{v}_k^{\,l})} \left\| \left(R_i \vec{v}_i^{\,j} + \vec{t}_i \right) - \left(R_k \vec{v}_k^{\,l} + \vec{t}_k \right) \right\|^2 ,$$

where $\vec{v}_i^{\,j}$ and $\vec{v}_k^{\,l}$ are the corresponding vertices of the matings defined by M and $(R_i, t_i), (R_k, t_k)$ are the euclidean transformations (i.e positioning) of pieces p_i and p_k. Unfortunately, this is no simple linear least squares minimization, as the unknowns include rotation matrices and the transformations as a whole must satisfy the constraint that they are *identical for all vertices of the same piece*. As such, this optimization problem defies analytical solutions and we therefore resort to tools from other disciplines, and in particular we propose to abstract the rearrangement problem as a *multi-body spring-mass system*, where the pieces are rigid 2D bodies with uniform density (and therefore with mass that is proportional to their area) and the vertices of the (given) mates are connected by springs of zero length and constant elasticity (i.e., having identical *spring constants*). The potential energy of such a spring-mass system is $U(x) = \sum_l \frac{1}{2} k x_l^2$, where x_l is the displacement from equilibrium length of spring l, and thus is identical to our objective function. We therefore apply numerical methods for multi-body spring-mass systems, while the initial pose (position and rotation) of each piece is chosen randomly inside the arena. The physical system is then set loose and with some damping (i.e., loss of energy due to friction) it converges to its minimal energetic state, as illustrated in Fig. 4A.

In practice there are off-the-shelf tools to solve the above system numerically, practically simulating the dynamical process that the system undergoes from initial condition until convergence. Here we use the Box2D physics engine [6], let it run while allowing the pieces to overlap, and upon convergence restart the process, this time while forbidding such overlaps. The end result is our solution (Fig. 4B).

6.2. Solving noisy puzzles with *unknown* matings

Let $P = \{p_1, \dots p_n\}$ be the set of puzzle pieces and let ε denote the noise level. We now seek the correct matings $M = \{m_1, \dots, m_{|M|}\}$ between the edges *and* the geometrical transformation of each piece. To do so, we develop a modified version of the hierarchical loops scheme [39], where the mass-spring minimization method from Sec. 6.1 is used to score the loops based on its success to position the pieces properly.

6.2.1 Hierarchical layered loops

As is usually done in puzzle solvers, we start by finding candidate mates for each edge, in our case by aggregating the set of all unordered pairs $\tilde{M} = \left\{ \{e_i^j, e_k^l\} \;\middle|\; e_i^j, e_k^l \in E \wedge \tilde{C}_1\left(e_i^j, e_k^l\right) \wedge \tilde{C}_2\left(e_i^j, e_k^l\right) \right\}$. Clearly, $\tilde{M}$ grows with higher noise levels (cf. Sec. 5).

As mentioned earlier, in crossing cuts puzzles with uniformly distributed random cuts, the probability of more than two cuts to meet at a point is nil (cf. Sec. 3). It directly follows that all *inner* puzzle junctions constitute exactly four pieces. We utilize this property to identify ordered lists of 4 mating candidates that form such junctions, or *loops*, as illustrated in Fig.5. Formally, a loop in the *clockwise* direction is a 4-tuple $(m_1, m_2, m_3, m_4) = \left(\left\{ e_A^{j_A}, e_B^{i_B} \right\}, \left\{ e_B^{j_B}, e_C^{i_C} \right\}, \left\{ e_C^{j_C}, e_D^{i_D} \right\}, \left\{ e_D^{j_D}, e_A^{i_A} \right\} \right)$ such that $m_k \in \tilde{M}$ and the following conditions hold:

- No piece appears twice, i.e. $p_A \neq p_B \neq p_C \neq p_D$.
- Each two consecutive matings that "enter" a piece p though its e_p^j edge, "exist" the same piece through an adjacent edge $e_p^{(j-1) \mod N_p}$, where N_p is the number of p's edges (and also vertices). In other words, it "exits" through an edge immediately *counterclockwise* to e_p^j along the piece border (e.g., edges e_B^4 and e_B^3 in Fig. 5B).
- The loop begins and ends with the same piece. This is true by definition as both the first and last matings contain the same edge of piece p_A

Since these basic loops are the building blocks for the puzzle reconstruction, and no piece can be missed, we

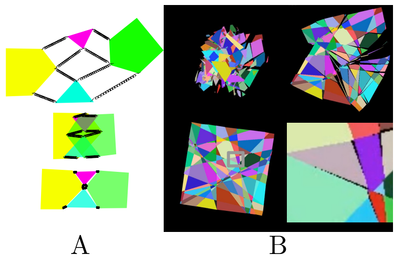


Figure 4: The abstraction of the puzzle problem as a multi-body spring-mass system. **A:** The puzzle with given matings is abstracted as a spring-mass system evolving over time. If the pieces are far apart, the springs pull them closer. When then pieces overlap, the springs pull them apart again. With some damping (i.e., loss of energy due to friction), the system eventually converges to minimize the total potential energy of the springs. **B:** Several snapshots of the simulation for a puzzle of 25 cuts, 940 pieces, and noise level being 2% of the box size. The bottom right cell is a closeup to show the approximated placement due to the noise. See animated examples in the Supp.

search for them exhaustively among all $O\left(|\tilde{M}|^4\right)$ possible mating 4-tuples, keeping only those that satisfy all of the above constraints.

Let $\mathcal{L}$ be the bag of basic loops computed as above. We now exploit partial *overlaps* between loops to identify *correct* matings more robustly instead of relying on $\tilde{M}$ matings alone. More specifically, the next stage of the puzzle reconstruction algorithm is searching for "higher order" loops, i.e., loops of loops, or *hierarchical loops* [39]. Denoting the basic 4-tuple loops in $\mathcal{L}$ as 0-loops, we now seek all possible x-loops by trying to enclose $(x-1)$-loops with partially overlapping 0-loops, as illustrated in Fig. 5C. Toward that end, let $(e_1, e_2 \dots e_k)$ be the list of edges along the boundary of some $(x-1)$-loop. For example, the boundary of the 0-loop in Fig. 5B is $(e_A^0, e_B^0, e_B^1, e_B^2, e_C^0, e_D^3, e_D^3, e_D^4, e_D^5)$. Starting with e_1 and ending with e_k, we progressively construct a higher level x-loop by searching and merging a proper 0-loop from $\mathcal{L}$ that matches a *sub-loop* of the current x-loop around e_i. For example, if we start from the boundary edge e_A^0 in Fig. 5B, we look for 0-loops that not only include that edge but also include the mating $\{e_B^4, e_A^1\}$. The loop that was found in this particular example constitutes $(\{e_A^0, e_F^0\}, \{e_F^3, e_G^1\}, \{e_G^0, e_B^0\}, \{e_B^4, e_A^1\})$ as shown in Fig. 5C. Typically, the new 0-loops found will need to match an existing sub-loop of 2 or 3 matings. If at some edge e_i more than a single 0-loop is found in $\mathcal{L}$, they too will be considered in order to store all possible x-loops that can enclose a given $(x-1)$-loop.

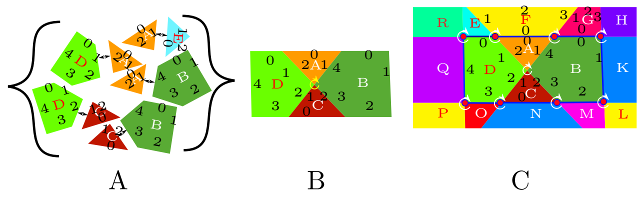


Figure 5: Loop formation from pairwise matings to the construction hierarchical loops. **A:** A bag $\tilde{M}$ of 5 potential matings, one of which (e_A^1, e_E^0) is wrong. **B:** The loop $(e_A^1, e_B^4) \rightarrow (e_B^3, e_C^2) \rightarrow (e_C^1, e_D^2) \rightarrow (e_D^1, e_A^2)$ is identified and supports the plausibility of its constituent matings. Note that the path ending with (e_A^1, e_E^0) does not close a loop because the mating (e_E^2, e_C^2) is not present in the bag. **C:** The border edges of the inner 0-loop are used one at a time to seek other partially overlapping 0-loops with the existing matings and pieces.

The process just described constructs the hierarchical loops in "layers" to produce a bag of x-loops for each layer x. Each of the 0-loops in $\mathcal{L}$ may produce several 1-loops, each of them may produce several 2-loops, and so forth. This process terminates at level $x_{\max}$ if not even a single $(x_{\max}+1)$-loop can be constructed, an event likely to happen if such loops overflow beyond the true puzzle boundary.

6.2.2 Ranking hierarchical loops

Although hierarchical loops require simultaneous consensus between growing numbers of participating matings, and thereby reduce significantly the possibility of wrong combinations, false positives are still possible. To rank better and worse loops, we utilize the fact that each of them is a small noisy puzzle of pieces P_{loop} and (known) matings M_{loop} (cf. Sec. 6.1),and that "correct" loops can be "solved" with little to no overlaps even when collisions are allowed (cf. Sec 6.1). We therefore employ the spring-mass mechanism and rank each x-loops by its convergence state. We first define the following "quality" measure

$$Q_{overlap} = \sum_{p_i \in P_{loop}} \frac{\left| A(p_i) \cap \left(\bigcup_{p_j \neq p_i} A(p_j) \right) \right|}{|A(p_i)|} \tag{1}$$

where $A(p_i)$ represents the *region* of piece p_i and the measure as whole is a modified Dice coefficient [12] between each piece and the rest of the pieces. Since the distance between all adjacent vertices in "correct" loops also must be small, we also consider the distances between corresponding vertices as defined by M_{loop} measured *after* collisions are prohibited:

$$Q_{dist} = \sum_{\vec{v}_i, \vec{v}_{i'}} \|\vec{v}_i - \vec{v}_{i'}\|^2 \tag{2}$$

Combining both scores into one rank we get

$$Q = w_1 \cdot Q_{overlap} + w_2 \cdot Q_{dist} \tag{3}$$

In our evaluation $w_1 = w_2 = 1$ obtained excellent results.

6.2.3 Merging hierarchical loops:

Even with the best hierarchical loop found at the maximum level, the process of puzzle reconstruction is not yet finished since the maximum level of hierarchical loops usually does not cover the entire puzzle. To complete the process and obtain the matings for the complete puzzle we now attempt to merge hierarchical loops. The x-loops are first sorted at each level x according to their rank, and this list is then scanned from the best and highest level loops.

More formally, let P_{agg}, M_{agg} denote the pieces and matings of the merging (or aggregation) process, initialized to be the best x_{max}-loop. For each $x = x_{max} \dots x_0$ scanning the sorted list of all x-loops, each is merged into the aggregated structure if several conditions hold. Assuming the pieces of the current x-loop under consideration are P_{loop}, and they are connected with M_{loop} matings, this loop is merged into P_{agg}, M_{agg} if

- at least one piece is shared with the aggregated structure, i.e $P_{agg} \cap P_{loop} \neq \emptyset$,
- at least one piece is novel, i.e $P_{agg} \cup P_{loop} \neq P_{agg}$,
- and there is no contradiction between the matings in M_{agg} and M_{loop}, i.e. if $\{e_A^i, e_B^j\} \in M_{loop}$ then either $\{e_A^i, e_B^j\} \in M_{agg}$ *or* none of the matings in M_{agg} contain edges e_A^i or e_B^j.

The merging process continues through the lowest ranked 0-loop, and is then repeated from the start until M_{agg} no longer changes during a full scan. This process must converge since the aggregation can include each possible mating at most once.

After the aggregated structure converges, the multi-body spring-mass process is performed one last time to position all the pieces P_{agg} properly based on the obtained mating M_{agg}. The result is the reconstructed crossing cut puzzle. The Supplementary Materials present several results, including full animated visualizations of the process.

7. *Pictorial* crossing cut puzzles

As the geometrical noise increases, the number of potential mates found using geometrical constraints (i.e $\tilde{C}_1$ and $\tilde{C}_2$) grows rapidly with the number of cuts (cf. Sec. 5). In these cases, using the pictorial content of the pieces can provide a big advantage. In particular, while the initial set of potential matings is obtained using geometrical constraints, ranking them based on pictorial content can drastically reduce admissible matings and thus the computational effort of the reconstruction algorithm from Sec. 6.2. Moreover, as real visual puzzles, as well as those studied in the literature, often incorporate pictorial component (cf. Sec. 1), it begs to consider this variation for crossing cuts puzzles also, thereby making them highly relevant for computational and machine vision as well. A typical pictorial noisy crossing cut puzzle is depicted in Fig. 6A.

While not easily observed at the scale of the figure, the geometric noise distances the available pictorial content of neighboring edges and thus complicates the way we can use it to determine their compatibility as plausible mates. Thus, we score a candidate mating by extrapolating [9] the information of the two corresponding puzzle pieces to a spatial band beyond their boundaries and obtain "dilated" pictorial pieces [24]. Matings are then scored as described in Fig 6, and only the best T matings are kept for further considerations for each edge (where T is a parameter).

8. Experimental Results

For evaluation, and without prior work on crossing cuts puzzles, we focused on formulation of performance metrics and reporting qualitative and quantitative results on a novel benchmark dataset. For the latter we synthesized both pictorial and apictorial random puzzles with varying global shape, number of crossing cuts, and level of noise. Results of the baseline algorithm for "clean" puzzles are not reported as it always provides perfect performance. The complete dataset is described in the Supp and open to the community [18].

We first tested our approach for puzzles with known matings to evaluate the degree to which the abstraction as a multi-body spring-mass systems (Sec. 6.1) provides desired results. Recall that under this scenario the input constitutes the noisy puzzle pieces $\tilde{P}$ and the ground truth matings M_{gt}, while the output is the euclidean transformation of each piece $(R_1, t_1), \ldots (R_n, t_n)$. The quantitative evaluation of such output is not straight forward, though, since qualitatively perfect solutions by the spring-mass system may differ by a global Euclidean transformation due to arbitrary choice of coordinate system in the representation of the pieces (cf. Sec. 2 and Fig. 1B). For this reason, we first globally align the obtained solution with the ground truth before comparing the placement of individual pieces. This is done by employing SVD for Least-Squares Rigid Motion [41] to find the global Euclidean transformation (R, t) that minimizes $\sum_{i=1}^{n} \sum_{j=1}^{N_i} w_i \|(R\overrightarrow{v}_i^j + t) - \overrightarrow{u}_i^j\|^2$ with $w_i = \frac{|A(p_i)|}{\sum_{k=1}^{n} |A(p_k)|}$, where u_i^j are the vertices of piece p_i in the ground truth and $\overrightarrow{v}_i^j$ are the corresponding vertices of $\tilde{p}_i$ in the obtained solution. The weights w_i are set proportional to the area of each piece to reflect on the greater importance of larger pieces on the puzzle shape.

Once the obtained global transform is applied to all pieces, it becomes possible to score the collective placement of the pieces in the solution. Unfortunately, Eq. 8 cannot be normalized to some canonical range (say $[0, 1]$) and thus is inconvenient for such scoring. We therefore consider the degree of area overlaps between the pieces in the solution vs. their ground truth counterpart. With proper weighting by piece size this yields the following measure

$$Q_{positions} = \sum_{i=1}^{n} w_i \cdot \frac{|A(p_i) \cap A(R_i \tilde{p}_i + t_i)|)}{|A(\tilde{p}_i)|}. \qquad (4)$$

Fig. 7A presents the results, with various noise levels that exceed half the average edge length.

With a system to evaluate the solutions by the multi-body spring-mass system established, we turn to evaluate the full algorithmic pipeline under *unknown* matings (Sec. 6.2), where the input is just the noisy pieces $\tilde{P}$ (and the bound on the noise level ξ) while the output includes the matings M and the Euclidean transformations for each piece. We seek to evaluate

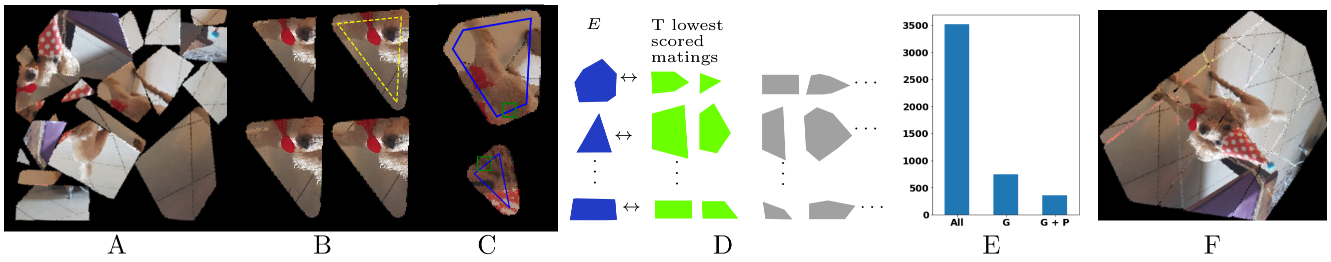


Figure 6: The stages of pictorial crossing cuts puzzle reconstruction. **A:** A pictorial crossing cuts puzzle is an unordered set of *pictorial*, noisy pieces. Recall that the noise is geometrical, not pictorial. **B:** Border extrapolation [9] is performed for each piece of the puzzle. Ordered left to right and top to bottom: (1) An original (geometrically noisy) piece from A. (2) The extrapolated piece for extrapolation radius of $2\varepsilon = 10$ pixels. The yellow polygon depicts the boundaries of the original piece, (3) The extrapolated piece without the original borders, (4) the same region taken from the original image. **C:** To compute the dissimilarity score $S(m)$ for each potential mating $m = \{e_i^j, e_k^l\}$, two running windows of size $2\varepsilon \times 2\varepsilon$ scan the two edges e_i^j, e_k^l *synchronously*, capturing both raw and extrapolated content of pieces p_i, p_k. The difference of the mean color value of all window pairs generates a vector, whose L_1 norm serves as $S(m)$. Note that the running windows cannot be completely synchronized in relative position since the edges may have different noisy lengths. However, the low pass filtering embedded in $S(m)$ renders it useful after all. **D:** Only the best T matings (in green) are kept for each edge, with T defined based on computational or time resources. **E:** The number of all matings combinations, potential matings obeying geometrical constraints (G) and admissible matings satisfying both geometrical and pictorial constraints (G+P) in the shown puzzle. Note how the pictorial constrain reduces the number by a factor of 4. **F:** Once the pictorially constrained set $\tilde{M}_p$ is obtained, the reconstruction proceeds exactly as described for the apictorial case (Sec. 6.2), in this case to perfection.

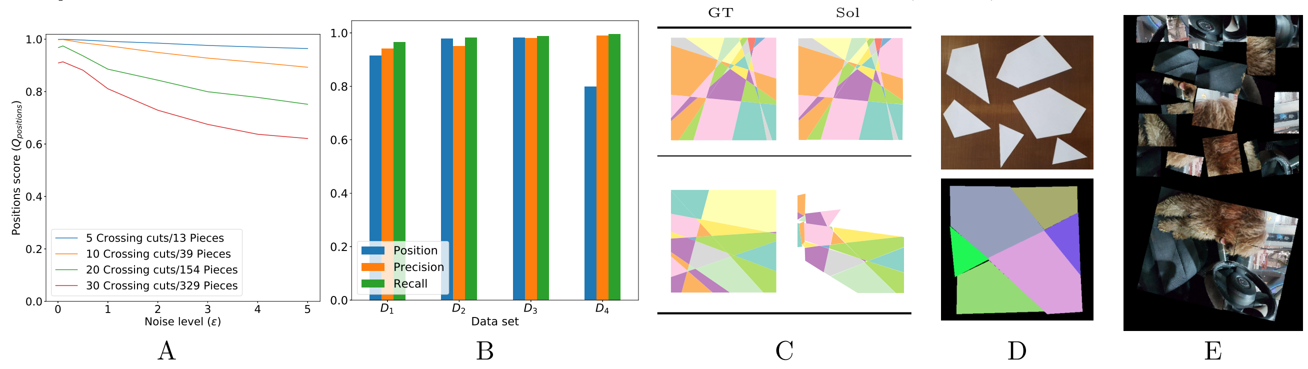


Figure 7: Experimental evaluation. **A:** Average score of piece positioning for known matings as a function of noise level, computed from 10 random puzzles for selected numbers of cuts/pieces. **C:** The reconstruction results on 4 datasets of 10 random puzzles each, showing the positioning score, and the precision and recall of the matings. The datasets evaluated are D_1:(a=8, p=26, ξ=1%, $\bar{\xi}$=10%), D_2:(a=10, p=39, ξ=0.5%, $\bar{\xi}$=6%), D_3:(a=19, p=131, ξ=0.1%, $\bar{\xi}$=2%), D_4:(a=35, p=435, ξ=0.01%, $\bar{\xi}$=0.4%), where a is number of cuts, p is average number of pieces (rounded), and $\xi, \bar{\xi}$ are noise levels relative to the diameter and average edge length, respectively. On average puzzles in datasets 1-3 took few seconds to solve on our test machine (Intel Core i5 3.2GHz 8G RAM), while those form dataset 4 took about a minute . **D:** A physical puzzle scanned by a camera and then solved by the crossing cuts algorithm. **E:** A (perfect) solution of a pictorial crossing cuts puzzle obtained with perturbed grid cuts and $\bar{\xi}$=5% geometrical noise.

both aspects, and while the positioning (i.e, the Euclidean transformations) are evaluated as above, our evaluation metric for the computed matings is inspired by the Neighbor Comparison Metric used in the square jigsaw puzzle literature (e.g., [7,31,36] but modified to compute area-weighted versions of the precision and recall of the computed matings. Fig. 7B depicts these measures shows that the mechanism based on raw matings, hierarchical loops, and merging ranked loops, obtains excellent results. Fig 7C presents a visual example for a success and a failure of two reconstructions. Fig 7D presents a physical crossing cuts puzzle which we solved after scanning and processing its image to obtain a geometric description of the pieces.

Finally, we show two *pictorial* puzzles, both reconstructed perfectly. Fig. 6A,F show a typical crossing cuts puzzle, with noise level of $\xi = 4\%$,$\bar{\xi} = 14\%$ higher than any of the non pictorial puzzles we solved. Fig 7E shows a pictorial square jigsaw puzzle with perturbed grid cuts (and hence perturbed square pieces) with noise level of ξ=5%, $\bar{\xi}$=10%. With almost identical pieces, the geometrical constraints are now marginalized, entailing a $\tilde{M}$ set particularly excessive in size unless pictorial constraints are employed. The pictorial puzzles took approximately 10 minutes to solve, about 30 times faster than their apictorial counterparts.

Acknowledgments: We thank Roy Hershfinkel for help in the physical puzzle experiment. This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 964854, the Helmsley Charitable Trust through the ABC Robotics Initiative, and the Frankel Fund of the Computer Science Department at Ben-Gurion University.

References

[1] Nagesh Adluru, Xingwei Yang, and Longin Jan Latecki. Sequential monte carlo for maximum weight subgraphs with application to solving image jigsaw puzzles. *International journal of computer vision*, 112(3):319–341, 2015. 1

[2] Naif Alajlan. Solving square jigsaw puzzles using dynamic programming and the hungarian procedure. *American Journal of Applied Sciences*, 6(11):1941, 2009. 1

[3] Fernanda A Andaló, Gabriel Taubin, and Sione Goldenstein. Solving image puzzles with a simple quadratic programming formulation. In *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 63–70. IEEE, 2012. 1

[4] Susana Brandão and Manuel Marques. Hot tiles: A heat diffusion based descriptor for automatic tile panel assembly. In *European Conference on Computer Vision*, pages 768–782. Springer, 2016. 1

[5] Horst Bunke and Guido Kaufmann. Jigsaw puzzle solving using approximate string matching and best-first search. In *International Conference on Computer Analysis of Images and Patterns*, pages 299–308. Springer, 1993. 1

[6] Erin Catto. Box2d. https://github.com/erincatto/Box2D. 5

[7] Taeg Sang Cho, Shai Avidan, and William T Freeman. A probabilistic image jigsaw puzzle solver. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 183–190. IEEE, 2010. 1, 8

[8] Min Gyo Chung, Margaret M Fleck, and David A Forsyth. Jigsaw puzzle solver using shape and color. In *ICSP'98. 1998 Fourth International Conference on Signal Processing (Cat. No. 98TH8344)*, volume 2, pages 877–880. IEEE, 1998. 1

[9] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004. 7, 8

[10] Johan De Bock, R De Smet, Wilfried Philips, and Johan D'Haeyer. Constructing the topological solution of jigsaw puzzles. In *2004 International Conference on Image Processing, 2004. ICIP'04.*, volume 3, pages 2127–2130. IEEE, 2004. 1

[11] Erik D. Demaine and Martin L. Demaine. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, 23(1):195–208, Jun 2007. 1

[12] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. 6

[13] Ni Fei, Fu Zhuang, Liu Renqiang, Cao Qixin, and Zhao Yanzheng. An image processing approach for jigsaw puzzle assembly. *Assembly Automation*, 27(1):25–30, 2007. 1

[14] Herbert Freeman and L Garder. Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Transactions on Electronic Computers*, (2):118–127, 1964. 1, 2

[15] Andrew C Gallagher. Jigsaw puzzles with pieces of unknown orientation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 382–389. IEEE, 2012. 1

[16] David Goldberg, Christopher Malon, and Marshall Bern. A global approach to automatic solution of jigsaw puzzles. In *Proceedings of the eighteenth annual symposium on Computational geometry*, pages 82–87. ACM, 2002. 1

[17] Shir Gur and Ohad Ben-Shahar. From square pieces to brick walls: The next challenge in solving jigsaw puzzles. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4029–4037, 2017. 1

[18] Peleg Harel. Computational polygonal puzzle datasets. http://icvl.cs.bgu.ac.il/polygonal-puzzle-solving/. 7

[19] David Koller and Marc Levoy. Computer-aided reconstruction and new matches in the forma urbis romae. *Bullettino Della Commissione Archeologica Comunale di Roma*, 2:103–125, 2006. 1

[20] Weixin Kong and Benjamin B Kimia. On solving 2d and 3d puzzles using curve matching. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–II. IEEE, 2001. 1, 2

[21] David A Kosiba, Pierre M Devaux, Sanjay Balasubramanian, Tarak L Gandhi, and K Kasturi. An automatic jigsaw puzzle solver. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 616–618. IEEE, 1994. 1

[22] Canyu Le and Xin Li. Jigsawnet: Shredded image reassembly using convolutional neural network and loop-based composition. *IEEE Transactions on Image Processing*, 2019. 1, 2

[23] Huei-Yung Lin and Wen-Cheng Fan-Chiang. Reconstruction of shredded document based on image feature matching. *Expert Systems with Applications*, 39(3):3324–3332, 2012. 1

[24] Hairong Liu, Shengjiao Cao, and Shuicheng Yan. Automated assembly of shredded pieces from multiple photos. *IEEE Transactions on Multimedia*, 13(5):1154–1162, 2011. 1, 2, 7

[25] Michael Makridis and Nikos Papamarkos. A new technique for solving a jigsaw puzzle. In *2006 International Conference on Image Processing*, pages 2001–2004. IEEE, 2006. 1, 2

[26] William Marande and Gertraud Burger. Mitochondrial dna as a genomic jigsaw puzzle. *Science*, 318(5849):415–415, 2007. 1

[27] Debajyoti Mondal, Yang Wang, and Stephane Durocher. Robust solvers for square jigsaw puzzles. In *2013 International Conference on Computer and Robot Vision*, pages 249–256. IEEE, 2013. 1

[28] Takenori Murakami, Fubito Toyama, Kenji Shoji, and Juichi Miyamichi. Assembly of puzzles by connecting between blocks. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008. 1

[29] Ture R Nielsen, Peter Drewsen, and Klaus Hansen. Solving jigsaw puzzles using image features. *Pattern Recognition Letters*, 29(14):1924–1933, 2008. 1

[30] Genady Paikin and Ayellet Tal. Solving multiple square jigsaw puzzles with missing pieces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4832–4839, 2015. 1

[31] Dolev Pomeranz, Michal Shemesh, and Ohad Ben-Shahar. A fully automated greedy square jigsaw puzzle solver. In *CVPR 2011*, pages 9–16. IEEE, 2011. 1, 2, 8

[32] Gerald M Radack and Norman I Badler. Jigsaw puzzle matching using a boundary-centered polar encoding. *Computer Graphics and Image Processing*, 19(1):1–17, 1982. 1, 2

[33] Daniel Rika, Dror Sholomon, Eli Omid David, and Nathan S Netanyahu. A novel hybrid scheme using genetic algorithms and deep learning for the reconstruction of portuguese tile panels. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1319–1327. ACM, 2019. 1

[34] Mahmut Şamil Sağıroğlu and Aytül Erçil. Optimization for automated assembly of puzzles. *Top*, 18(2):321–338, 2010. 1, 2

[35] Hijung Shin, Christos Doumas, Thomas Funkhouser, Szymon Rusinkiewicz, Kenneth Steiglitz, Andreas Vlachopoulos, and Tim Weyrich. Analyzing and simulating fracture patterns of theran wall paintings. *Journal on Computing and Cultural Heritage (JOCCH)*, 5(3):10, 2012. 2

[36] Dror Sholomon, Omid David, and Nathan S Netanyahu. A genetic algorithm-based solver for very large jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1767–1774, 2013. 1, 8

[37] Dror Sholomon, Omid E David, and Nathan S Netanyahu. A generalized genetic algorithm-based solver for very large jigsaw puzzles of complex types. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014. 1

[38] Kilho Son, James Hays, and David B Cooper. Solving square jigsaw puzzles with loop constraints. In *European Conference on Computer Vision*, pages 32–46. Springer, 2014. 1, 2

[39] Kilho Son, James Hays, and David B Cooper. Solving square jigsaw puzzle by hierarchical loop constraints. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 1, 5, 6

[40] Kilho Son, James Hays, David B Cooper, et al. Solving small-piece jigsaw puzzles by growing consensus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1193–1201, 2016. 1

[41] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *Computing*, 1(1), 2017. 7

[42] Fubito Toyama, Yukihiro Fujiki, Kenji Shoji, and Juichi Miyamichi. Assembly of puzzles using a genetic algorithm. In *Object recognition supported by user interaction for service robots*, volume 4, pages 389–392. IEEE, 2002. 1

[43] Efthymia Tsamoura and Ioannis Pitas. Automatic color based reassembly of fragmented images and paintings. *IEEE Transactions on Image Processing*, 19(3):680–690, 2009. 1, 2

[44] Roger W Webster, Paul S LaFollette, and Robert L Stafford. Isthmus critical points for solving jigsaw puzzles in computer vision. *IEEE transactions on systems, man, and cybernetics*, 21(5):1271–1278, 1991. 1

[45] Haim Wolfson, Edith Schonberg, Alan Kalvin, and Yehezkel Lamdan. Solving jigsaw puzzles by computer. *Annals of Operations Research*, 12(1):51–64, 1988. 1

[46] Xingwei Yang, Nagesh Adluru, and Longin Jan Latecki. Particle filter with state permutations for solving image jigsaw puzzles. In *CVPR 2011*, pages 2873–2880. IEEE, 2011. 1

[47] Feng-Hui Yao and Gui-Feng Shao. A shape and image merging technique to solve jigsaw puzzles. *Pattern Recognition Letters*, 24(12):1819–1835, 2003. 1

[48] Rui Yu, Chris Russell, and Lourdes Agapito. Solving jigsaw puzzles with linear programming. *arXiv preprint arXiv:1511.04472*, 2015. 1

[49] Kang Zhang and Xin Li. A graph-based optimization algorithm for fragmented image reassembly. *Graphical Models*, 76(5):484–495, 2014. 1, 2

[50] Yu-Xiang Zhao, Mu-Chun Su, Zhong-Lie Chou, and Jonathan Lee. A puzzle solver and its application in speech descrambling. In *WSEAS International Conference on Computer Engineering and Applications*, pages 171–176, 2007. 1